Divide and Conquer: Efficient Density-Based Tracking of 3D Sensors in Manhattan Worlds

Yi Zhou [1,2]*, Laurent Kneip [1,2], Cristian Rodriguez [1,2], Hongdong Li [1,2]

[1] Research School of Engineering, The Australian National University
[2] Australian Centre for Robotic Vision
{yi.zhou, laurent.kneip, cristian.rodriguez, hongdong.li}@anu.edu.au

Abstract. 3D depth sensors such as LIDARs and RGB-D cameras have become a popular choice for indoor localization and mapping. However, due to the lack of direct frame-to-frame correspondences, the tracking traditionally relies on the iterative closest point technique which does not scale well with the number of points. In this paper, we build on top of more recent and efficient density distribution alignment methods, and notably push the idea towards a highly efficient and reliable solution for full 6DoF motion estimation with only depth information. We propose a divide-and-conquer technique during which the estimation of the rotation and the three degrees of freedom of the translation are all decoupled from one another. The rotation is estimated absolutely and drift-free by exploiting the orthogonal structure in man-made environments. The underlying algorithm is an efficient extension of the mean-shift paradigm to manifold-constrained multiple-mode tracking. Dedicated projections subsequently enable the estimation of the translation through three simple 1D density alignment steps that can be executed in parallel. An extensive evaluation on both simulated and publicly available real datasets comparing several existing methods demonstrates outstanding performance at low computational cost.

1 Introduction

3D depth sensors are a powerful alternative to cameras when it comes to automated localization and mapping. They perform especially well in man-made indoor environments, which are often composed of homogeneously colored planar pieces, and thus provide sufficient well-defined 3D structures for depth sensors, but insufficient texture for a reliable application of classical image-based localization techniques. Further advantages of active sensing are given by absolute (metric) scale operation (and therefore absence of scale drift) and resilience against illumination or appearance changes in the environment, ultimately even permitting operation in complete darkness. Depth sensors are an engineering answer to the inverse problem of structure-from-motion, and ubiquitous success is demonstrated by numerous successful applications in robotics [1, 2], autonomous

[*] Corresponding author.

driving (e.g. *Google Chauffeur*), and—more recently—consumer electronics (e.g. *Google Tango, Meta Glass*).

Depth sensors produce point cloud measurements. The fundamental problem behind incremental motion estimation with depth sensors therefore is the registration of two 3D point sets A and B. The most popular technique by far is given by the Iterative Closest Point (ICP) method [3]. The basic idea is straightforward: We find approximate correspondences between pairs of points between A and B by simply associating the spatially nearest neighbor of set B to each point of set A. We then minimize the sum of squared distances over a euclidean transformation in closed form. We finally iterate over these two steps until convergence. The complexity of the algorithm is an immediate consequence of the need to find the closest point for each point in each iteration. Even the fastest implementations [4, 5] therefore fail to deliver real-time performance on CPU as soon as we consider modern sensors returning dense depth images at VGA resolution. Distance-transform based ICP variants such as the ones used in KinectFusion [6] and Kintinuous [7] achieve real-time performance, however only by leveraging the power of a GPU.

A more efficient alternative registration principle transforms the data into lower dimensional, spatial density distribution functions [8]. The general advantage of density alignment based methods is that they do no longer depend on the establishment of one-to-one or even weighted, fuzzy one-to-many point correspondences [9]. Our work lifts this concept to a general, real-time motion estimation framework for 3D sensors. The key of our approach consists of exploiting the structure of man-made environments, which often contain sets of orthogonal planar pieces. We furthermore rely on efficient dense surface normal vector computation in order to estimate the rotation independently of the translation. As we will show, the exploitation of this prior furthermore allows us to split up the translational alignment of the density distribution functions into three independent steps, namely one along each direction in the corresponding cartesian coordinate frame.

In summary, we present a highly efficient motion estimation framework for popular 3D sensors such as the Microsoft Kinect, based on alignment of density distribution functions. Our contributions are listed as follows:

- Efficient, decoupled estimation of camera rotation using mean-shift for multi-mode tracking in surface normal vector distributions.
- Estimation of absolute rotation by exploiting the properties of Manhattan Worlds, thus resulting in manifold-constrained multi-mode tracking.
- Efficient decoupled estimation of individual translational degrees of freedom through 1D kernel density estimates.
- Integration into a real-time framework able to process dense depth images with VGA resolution at more than 50Hz on a laptop with only CPU resources. The result is an attractive 6 DoF tracker for autonomous mobile systems, which often have limited computational resources or energy supply.

We conclude the introduction by reviewing related work. Section 2 then introduces our main idea for motion estimation in Manhattan Worlds based on

3D sensors. The decoupled estimation of rotation and translation are presented in Sections 3 and 4, respectively. Section 5 finally presents our extensive experimental evaluation on both simulated and real data. We test and evaluate our algorithm against existing alternatives on publicly available datasets, showcasing outstanding performance at the lowest computational cost.

Related work: 3D Point set registration is a traditional problem that has been investigated extensively in the computer vision community. We are limiting the discussion to methods that process mainly rigid, geometric information. The most commonly used method is given by the ICP algorithm [3], which performs registration through iterative minimization of the SSD distance between spatial neighbors in two point sets. The costly repetitive derivation of point-to-point correspondences can be circumvented by representing and aligning point clouds using density distribution functions. The idea goes back to [10] and [11], who represent point clouds as explicit Gaussian Mixture Models (GMM) or implicit Kernel Density Estimates (KDE), and then find the relative transformation (not necessarily Euclidean) by aligning those density distributions. [8] summarizes the idea of using GMMs for finding the aligning transformation, and notably derives a closed-form expression for computing the L2 distance between two GMMs. Yet another alternative which avoids the establishment of point-to-point correspondences is given by [12], which utilizes a distance transformation in order to efficiently and robustly compute the cost of an aligning transformation. The distance transformation itself, however, is again computationally intensive.

Classical ICP or even density alignment based methods are prone to local minima as soon as the displacement is too large. In order to tackle situations of large view-point changes, [13] investigated globally optimal solutions to the point set registration problem. This method is however inefficient and thus not suited for real-time applications, where the frame-to-frame displacement anyway remains small enough for a successful application of local methods.

From a more modern perspective, the ICP algorithm and its close derivatives [4–7] still represent the algorithm of choice for real-time LIDAR tracking. The upcoming of RGB-D cameras has however led to a new generation of 2D-3D registration algorithms that exercise a hybrid use of both depth and RGB information. [14] for instance uses the depth information along with the optimized relative transformation to warp the image from one frame to the next, thus permitting direct and dense photometric error minimization. [15–18] apply a similar idea to RGB camera tracking. More recently, [19] even applied ICP and distance transforms to semi-dense 2D-3D registration.

The special structure of man-made environments can be exploited to simplify or even robustify the formulation of motion estimation with exteroceptive sensors. [20] and [21] introduce planar surfaces into the mapper which are often contained in our man-made environments. [22] combines point and plane features towards fast and accurate 3D registration. In our work, we additionally exploit the fact that indoor environments such as corridors frequently contain orthogonal structure in the surface arrangement. [23] coined the term *Manhattan World* (MW) to denote such an environment, and they estimated the camera orienta-

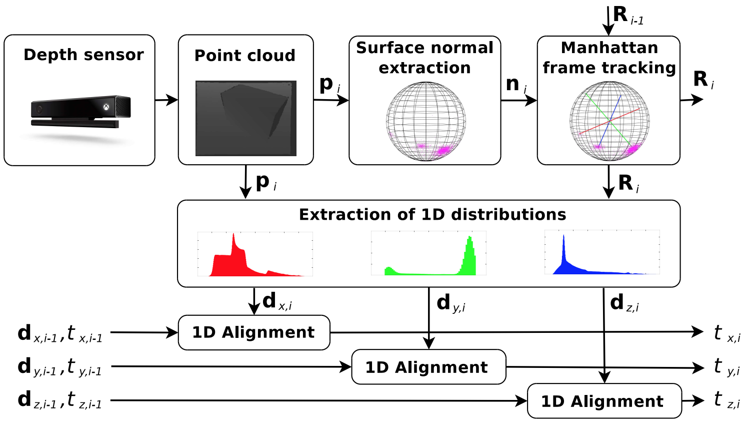


Fig. 1. Overview of the proposed, decoupled motion estimation framework for 3D sensors in Manhattan World.

tion through Bayesian vanishing point estimation in a single RGB image. [24] presents a video compass using a similar idea. Tracking the *Manhattan Frame* (MF) can be regarded as absolute orientation estimation, and thus leads to significant reduction or even complete elimination of the rotational drift. Silberman et al. [25] improve VP-based MW orientation estimation by introducing depth and surface normal information obtained from 3D sensors. More recently, [26] proposes the inference of an explicit probabilistic model to describe the world as a mixture of Manhattan frames. They employ an adaptive Markov-Chain Monte-Carlo sampling algorithm with Metropolis-Hasting split/merge moves to identify von-Mises-Fisher distributions of the surface normal vectors. In [27], they adapt the idea to a more computationally friendly approach for real-time tracking of a single, dominant MF. Our work is closely related, except that our mean-shift tracking scheme [28] is simpler and more computationally efficient than the MAP inference scheme presented in [27], which depends on approximations using the Karcher mean in order to achieve real-time performance. We furthermore extend the idea to full 6DoF motion estimation.

2 Overview of the Proposed Algorithm

Our method is summarized in Figure 1, and consists of three main steps. Note again that we use only depth information:

– We first start by extracting surface normal vectors $\mathbf{n}_i$ from the measured point clouds, which later allows us to compute the orientation of the sensor independently of the translation. Our method is a hyper-threaded CPU implementation of the approach presented in [29], which can efficiently return normal vectors for every pixel in a dense depth image. In order to get smooth and regularized surface normal vectors, the depth map is pre-processed by a smoothing guided filter [30].

- We then rely on the assumption that there is a dominant MF in the environment. This allows us to simply track a number of modes in the density distribution of the surface normal vectors, which can be done in a non-parametric way by employing the mean shift algorithm on the unit sphere. It prevents us from having to identify the parameters of a complete explicit model of the density distribution function. We present a manifold-constrained mean-shift algorithm that takes the orthogonality prior into account. Note that the optimization of the rotation is not a classical registration step, but a simple tracking procedure that uses information of a single frame only to produce a drift-free estimate of the absolute orientation.
- By knowing the absolute orientation in each frame, we can easily unrotate the point clouds of a frame pair and assume that the transformation that separates them is a pure translation. A further benefit is that the principal directions of a Gaussian Mixture Model of the point cloud can be constrained to align with the basis axes. In other words, the covariance matrices become diagonal by which the purely translational alignment cost can effectively be split up into three independent terms, namely one for each dimension. We are therefore allowed to simply solve for each translational degree of freedom independently. We notably do so by extracting kernel density distributions of the point clouds projected onto the basis axes, and by performing three simple 1D alignments. Again note that—due to the unrotation—the obtained relative displacement is immediately expressed in the world frame.

We will in the following explain the details of the rotation and translation alignment.

3 Absolute Orientation Estimation Based on Manifold-Constrained Mean-Shift Tracking

We estimate the absolute orientation by tracking a dominant MF in the surface normal vector distribution of each frame. We will start by introducing the mean-shift tracking scheme that operates under the assumption that a sufficiently close initialization point is known. We then conclude by explaining the initialization in the very first frame, which builds on top of our mean-shift extension.

3.1 Basic idea

For structures that obey the MW assumption, the surface normal vectors $\mathbf{n}_i$ have an organized distribution on the unit sphere $\mathbb{S}^2$, which can be exploited for recognizing the MW orientation. It is reasonable to assume that the unit vectors $\mathbf{n}_i$ are samples of a probability density function, as they are more likely to be distributed around the basis axes of the MW (in both directions). The process of finding the dominant axes is therefore equivalent to mode seeking in this density distribution (i.e. finding local maxima in the density distribution function). The modes are additionally constrained to be orthogonal with respect to each other.

We therefore express the MF by a proper 3D rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ of which each column $\mathbf{r}_j$ captures the direction of one of the dominant axes of the MF. Special care however needs to be taken in order to deal with the non-uniqueness of the representation, as each $\mathbf{r}_j$ could in principle be replaced by its negative (although we ensure that $\mathbf{R}$ always remains a right-handed matrix).

A popular, fast, and notably non-parametric method to seek modes is given by the mean shift algorithm [31]. Given an approximate location for a mode, the algorithm applies local Kernel Density Estimation (KDE) to iteratively take steps in the direction of increasing density. We apply this idea to our unit normal vectors on the manifold $\mathbb{S}^2$ using a Gaussian kernel over conic section windows of the unit sphere. The result is optimal under the assumption that the angles between the normal vectors and their corresponding mode centre have a Gaussian distribution. We independently compute one mean shift vector for each basis vector $\mathbf{r}_j$, which potentially results in a non-orthogonal updated MF $\hat{\mathbf{R}}$. We therefore finish each overall iteration by reprojecting $\hat{\mathbf{R}}$ onto the nearest $\mathbf{R} \in \mathrm{SO}(3)$. The following explains the update of each mode within a single mean-shift iteration, as well as the projection back onto SO(3).

3.2 Mean shift on the unit sphere

The core of our method is a single mean shift iteration for a dominant axis given a set of normal vectors on $\mathbb{S}^2$. It works as follows:

- We start by finding all normal vectors that are within a neighbourhood of the considered centre $\mathbf{r}_j$. The extent of this neighbourhood is notably defined by the kernel-width of our KDE. In our case, the window is a conic section of the unit sphere and the apex angle of the cone θ_{window} defines the size of the local window. Relevant normal vectors for mode j need to lie inside the respective cone, and thus satisfy the condition

$$\|\mathbf{n}_i \times \mathbf{r}_j\| < \sin(\frac{\theta_{\text{window}}}{2}). \tag{1}$$

  Let us define the index i_j which iterates through all $\mathbf{n}_i$ that fulfill the above condition. Note that—if choosing $\theta_{\text{window}} < \frac{\pi}{2}$—every $\mathbf{n}_i$ contributes to at most one mode.
- We then project all contributing $\mathbf{n}_{i_j}$ into the tangential plane at $\mathbf{r}_j$ in order to compute a mean shift. Let

$$\mathbf{Q} = \left[\mathbf{r}_{mod(j+1,3)}\ \mathbf{r}_{mod(j+2,3)}\ \mathbf{r}_{mod(j+3,3)}\right]. \tag{2}$$

  Then

$$\mathbf{n'}_{i_j} = \mathbf{Q}^T \mathbf{n}_{i_j} \tag{3}$$

  represents the normal vector rotated into the MF, with a cyclic permutation of the coordinates such that the last coordinate is along the direction of axis j. In order for the distances in the tangential plane to represent proper geodesics on $\mathbb{S}^2$ (or equivalently angular deviations), we apply the Riemann

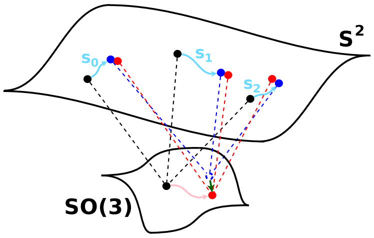


Fig. 2. Illustration of our cascaded manifold-constrained mean-shift implementation. We first compute updates $\mathbf{s}_j$ for each mode on $\mathbb{S}^2$, which brings us from the black to the blue modes. The blue modes however do no longer represent a point on the underlying manifold SO(3). We find the nearest rotation through a projection onto the manifold (green arrow), thus returning the red modes which are closest and at the same time fulfill the orthogonality constraint.

logarithmic map. The rescaled coordinates in the tangential plane are given by

$$\mathbf{m'}_{i_j} = \frac{\sin^{-1}(\lambda)\,\mathrm{sign}(n'_{i_j,z})}{\lambda} \begin{bmatrix} n'_{i_j,x} \\ n'_{i_j,y} \end{bmatrix}, \qquad (4)$$

$$\text{where } \lambda = \sqrt{n'^2_{i_j,x} + n'^2_{i_j,y}}.$$

Note that this projection has the advantage of correctly projecting normal vectors from either direction into the same tangential plane.

- We compute the mean shift in the tangential plane

$$\mathbf{s'}_j = \frac{\sum_{i_j} e^{-c\|\mathbf{m'}_{i_j}\|^2}\mathbf{m'}_{i_j}}{\sum_{i_j} e^{-c\|\mathbf{m'}_{i_j}\|^2}}. \qquad (5)$$

where c is a design parameter that defines the width of the kernel.

- To conclude, we transform the mean shift back onto the unit sphere using the Riemann exponential map

$$\mathbf{s}_j = \overline{\begin{bmatrix} \frac{\tan(\|\mathbf{s'}_j\|)}{\|\mathbf{s'}_j\|}\mathbf{s'}^T_j & 1 \end{bmatrix}}^T, \qquad (6)$$

where $\overline{[\cdot]}$ returns the input 3-vector divided by its norm.

- The updated direction $\hat{\mathbf{r}}_j$ is finally obtained by reapplying the current rotation with permuted axes

$$\hat{\mathbf{r}}_j = \mathbf{Q}\mathbf{s}_j. \qquad (7)$$

3.3 Maintaining orthogonality

After computing a mean shift for each mode $\mathbf{r}_j$, we effectively obtain an expression for the updated "rotation matrix"

$$\hat{\mathbf{R}} = \begin{bmatrix} \hat{\mathbf{r}}_0 & \hat{\mathbf{r}}_1 & \hat{\mathbf{r}}_2 \end{bmatrix}. \qquad (8)$$

This update may however violate the orthogonality constraint on our rotation matrix. We easily circumvent this problem by re-projecting $\hat{\mathbf{R}}$ onto the closest matrix on $SO(3)$ under the Frobenius norm. Each column of $\hat{\mathbf{R}}$ is re-weighted by a factor λ_i which describes how certain the observation of a direction is. In order to determine the weighting factors, we introduce a non-parametric variance approximation by utilizing a double parzen-widow-based KDE. The method is detailed in the supplemental material. The updated rotation matrix is finally given by

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T, \text{ where} \tag{9}$$

$$[\mathbf{U}, \mathbf{D}, \mathbf{V}] = \mathrm{SVD}(\begin{bmatrix}\lambda_0\hat{\mathbf{r}}_0 \ \lambda_1\hat{\mathbf{r}}_1 \ \lambda_2\hat{\mathbf{r}}_2\end{bmatrix}). \tag{10}$$

As illustrated in Figure 2, our method thus represents a double, cascaded manifold-constrained mean-shift extension, where the update of each mode is enforced to remain on the $\mathbb{S}^2$ manifold, and the combination of all three modes is each time enforced to remain an element on the $SO(3)$ manifold. In other words, in each iteration we compute the $SO(3)$-consistent update that is closest to the individual mean-shift updates.

3.4 Initialization in the first frame

We use mean-shift clustering to initialize the algorithm, and thus build on top of our MF tracking scheme. The procedure is summarized in Figure 3. We simply run the MF tracking procedure for 100 times, each time starting from a random initial rotation. This returns a redundant set of candidate MFs, within which we need to identify the most dominant cluster in order to complete the initialization. In fact, typically only a very small number of trials will not converge to the dominant MF if there is only one MF in the observed scene. However, the MF estimates are not directly comparable since one and the same MF may indeed be found or represented by any permutation or negation of individual basis vectors, as long as the result remains a right-handed matrix. In fact, there are 24 possible representations for one and the same MF. In order to render the results comparable and identify the dominant MF cluster, we convert the matrices into a canonical form based on a set of simple rules. For instance, the number of possible representations can already be reduced to 4 by simply requiring the basis vector with the potentially highest z-coordinate to be the one corresponding to the z-axis. To finally identify the dominant cluster, we simply group them based on a simple distance metric between rotation matrices, as well as a fixed threshold.

4 Translation Estimation through Separated 1-D Alignments

In this section, we show that by taking advantage of the MW properties, the translation in each dominant direction can be estimated separately. We then discuss the 1D alignments which rely on kernel density distribution functions. A convergence analysis is given in Section 2 of the supplementary material.

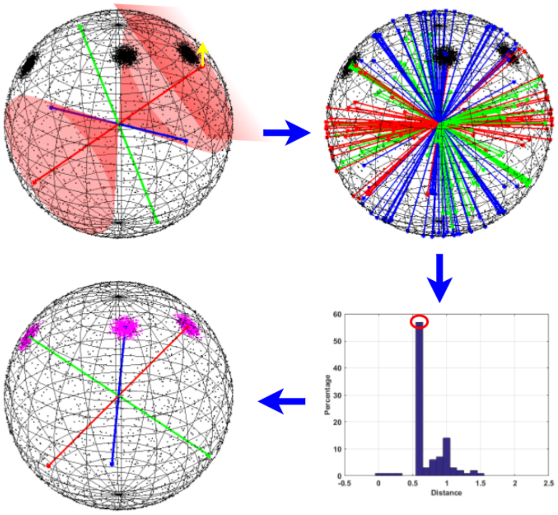


Fig. 3. The mechanism of the initial Manhattan frame seeking. The first figure shows a random initial MF. As indicated by one example, each dominant direction is refined by performing mean-shifts on the corresponding tangential plane. The second figure shows the redundant result obtained after full MF fitting from 100 random starts. The redundancy of the estimated rotation matrices **R** is removed by first converting them into a canonical form, and then performing histogram-based non-maximum suppression. The final result is shown in the fourth figure. For the sake of a clear visualization, the illustrated example is contaminated by a rather significant amount of uniformly distributed noise. Note that the proposed seeking strategy is even able to find multiple MFs in the environment, and thus come up with a mixture of Manhattan frames.

4.1 Independence of the three translational degrees of freedom

Although we are not using an explicit model for representing the density distributions, let us assume for a moment that it is given by a simple Gaussian (i.e. a toy GMM) to see the implications of a Manhattan world and a known absolute orientation of the Manhattan frame. A Gaussian in 3D with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$ is simply given by

$$\phi(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{\exp[-0.5(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})]}{\sqrt{(2\pi)^3|\det(\boldsymbol{\Sigma})|}}. \tag{11}$$

There are two Gaussians in two frames and—using the known absolute orientations to unrotate the point clouds—they are separated by a pure translation $\mathbf{t}$. By adding $\mathbf{t}$ to the mean of the Gaussian in the second frame, the kernel correlation between the two Gaussians can be calculated by

$$\begin{aligned} \mathrm{D} &= \int \phi(\mathbf{x}|\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1)\phi(\mathbf{x}|(\boldsymbol{\mu}_2+\mathbf{t}),\boldsymbol{\Sigma}_2)d\mathbf{x} \\ &= \phi(\mathbf{0}|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2-\mathbf{t},\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2). \end{aligned} \tag{12}$$

We now simplify the case by assuming that the unrotated point clouds can be expressed by a 3D Gaussian distribution with a diagonal covariance matrix. This is reasonable since the unrotated point clouds will indeed contain sets of points that are parallel to the basis axes. Let $\boldsymbol{\Sigma}_d = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 = \operatorname{diag}(\sigma_{dx}, \sigma_{dy}, \sigma_{dz})$, and $\boldsymbol{\mu}_d = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. Then the kernel correlation becomes

$$\begin{aligned}\mathrm{D} &= \frac{\exp[-0.5(\frac{(t_x-\mu_{dx})^2}{\sigma_{dx}} + \frac{(t_y-\mu_{dy})^2}{\sigma_{dy}} + \frac{(t_z-\mu_{dz})^2}{\sigma_{dz}})]}{\sqrt{(2\pi)^3\sigma_{dx}\sigma_{dy}\sigma_{dz}}} \\ &= k \cdot e^{\frac{(t_x-\mu_{dx})^2}{-2\sigma_{dx}}} e^{\frac{(t_y-\mu_{dy})^2}{-2\sigma_{dy}}} e^{\frac{(t_z-\mu_{dz})^2}{-2\sigma_{dz}}}. \end{aligned} \tag{13}$$

The goal of the alignment in this toy example is to find $\mathbf{t}$ such that D is maximized. It is clear that the above expression involves the product of three independent and positive elements, which means that maximizing each one independently will also maximize the overall distance between the Gaussians. Note that—in practice—the shape of the measured distributions is also influenced by occlusions under motion. However, we confirmed through our experiments that this has a neglible influence on the accuracy of the translation estimation in frame-to-frame motion estimation, as the location of the peaks in the distribution typically remains very stable.

4.2 Alignment of kernel density distributions

Our translation alignment procedure relies on implicit kernel density distribution functions. Assuming that the absolute orientation with respect to the MF is given, each degree of freedom can be solved independently, as in our toy GMM-based example. We therefore compensate for the absolute rotation of the point clouds, and project them onto each basis axis to obtain three independent 1D point sets. Inspired by popular point-set registration works, we then express the 1D point sets via kernel density distribution functions. We sample the function at regular intervals between the minimal and the maximal value. A Gaussian kernel with constant width is used to extract the density at each sampling position. Finally, the alignment between pairs of discretely sampled 1D signals seeks the 1D shift that minimizes the correlation distance between the two signals. It is worth to note that minimizing the correlation distance is equivalent to maximizing the kernel correlation as discussed above. The correlation distance for each pair of 1-D discrete signals is defined as

$$\mathcal{F} = \sum_{i=1}^{n} \left(f(x_i + t) - g(x_i)\right)^2, x_i \in X, \tag{14}$$

where X denotes a set of sampling positions for which a density is extracted using a Gaussian kernel. The functions f and g record the density at discrete sampling positions. The correlation distance is the sum over the squared differences at each sampling position. t is continuous, and we therefore obtain density values in between the sampled positions by employing linear interpolation. Note that the procedure has linear complexity in the number of points. The convergence analysis of the 1-D alignment is detailed in the supplemental material.

5 Experimental validation

This section evaluates our algorithm. We start by discussing parameter choices. We then compare our algorithm against two other established state-of-the-art motion estimation solutions on several publicly available datasets. We furthermore provide a reconstruction of a building-scale scene, and conclude by discussing the limitations and failure cases of our method.

Further simulation experiments and analyses are provided in the supplemental material. It contains 1) an evaluation of the robustness of our manifold-constraint mean-shift based MF-seeking strategy and 2) the benefit of aligning the point density distributions along the main axes of the MF.

5.1 Parameter configuration

In the initial MF seeking (i.e. the initialization of the absolute rotation from scratch), the total number of random starts N_{trial} is set to 100. The apex angle is set to 90° during the initialization and 20° during later tracking. This reduction of the cone apex angle is justified by the assumption that the orientation of the MF does not change too much under smooth motion. Each iterative mean-shift procedure terminates once the angle of the update rotation within one iteration falls below a threshold angle $\theta_{Converge}$, which we set to 1°. The factor c in Eq (5) is set to 20. Mean-shift updates are furthermore required to have a minimum number N_{min} of surface normal vectors within the dual-cone. The value of N_{min} depends on the resolution of the input depth map. For low resolution sensors (e.g. Kinect v.1, 160×120), $N_{min} = 30$. For high resolution sensors (Kinect v.2, 640×480), $N_{min} = 100$.

The parameters for the translation estimation contain two parts. The first part concerns the extraction of the density distributions. The sampling between the minimum and maximum value along each basis axis is made in constant intervals of $\delta_s = 0.01m$. The standard deviation σ of the Gaussian kernel for the KDEs is set to $0.03m$. The second part concerns the actual minimization of the correlation distance between each pair of 1D distributions. We simply employ gradient descent with an initial step size of 0.001m. The search range is furthermore restricted to $\pm 0.1m$.

5.2 Evaluation on real data

We compare the performance of our method against two state-of-the-art, open-source motion estimation implementations for 3D sensors, namely DVO [14] and KinectFusion's ICP [6, 7]. DVO uses both RGB images and depth maps while ICP and our algorithm use only depth information. We evaluate the methods on several recently published and challenging benchmark datasets from the TUM

Dataset	**DVO**				**ICP**				**Our Method**			
	$\hat{\mathbf{e}}_{\mathbf{R}}$	$\hat{\mathbf{e}}_{\mathbf{t}}$	$\tilde{\mathbf{e}}_{\mathbf{R}}$	$\tilde{\mathbf{e}}_{\mathbf{t}}$	$\hat{\mathbf{e}}_{\mathbf{R}}$	$\hat{\mathbf{e}}_{\mathbf{t}}$	$\tilde{\mathbf{e}}_{\mathbf{R}}$	$\tilde{\mathbf{e}}_{\mathbf{t}}$	$\hat{\mathbf{e}}_{\mathbf{R}}$	$\hat{\mathbf{e}}_{\mathbf{t}}$	$\tilde{\mathbf{e}}_{\mathbf{R}}$	$\tilde{\mathbf{e}}_{\mathbf{t}}$
TUM 1	4.91	0.15	4.46	0.13	6.64	0.17	6.01	0.15	**1.02**	**0.02**	**0.82**	**0.01**
TUM 2	2.21	0.10	1.59	0.06	9.07	0.27	7.57	0.26	**0.76**	**0.03**	**0.55**	**0.02**
TUM 3	10.90	0.20	3.89	0.07	12.80	0.17	10.17	0.16	**0.94**	**0.04**	**0.70**	**0.02**
TUM 4	**0.57**	**0.02**	**0.47**	**0.02**	8.66	0.29	7.17	0.27	1.01	0.03	0.80	0.03
TUM 5	**0.94**	**0.02**	**0.74**	**0.02**	16.80	0.24	14.19	0.22	1.12	0.04	0.87	**0.02**
IC 1	10.91	1.36	9.37	0.88	6.78	0.15	5.42	0.10	**1.55**	**0.13**	**1.12**	**0.09**
IC 2	6.97	0.70	6.58	0.45	6.31	0.16	5.28	0.10	**1.53**	**0.10**	**1.07**	**0.08**

Table 1. Performance comparison on several indoor datasets.

RGB-D [32] and IC-NUIM [33] series. The datasets we picked for evaluation are listed below and the results are summarized in Table 1. The selection of the datasets is based on the existence of sufficient MW structure in the observed scenes.

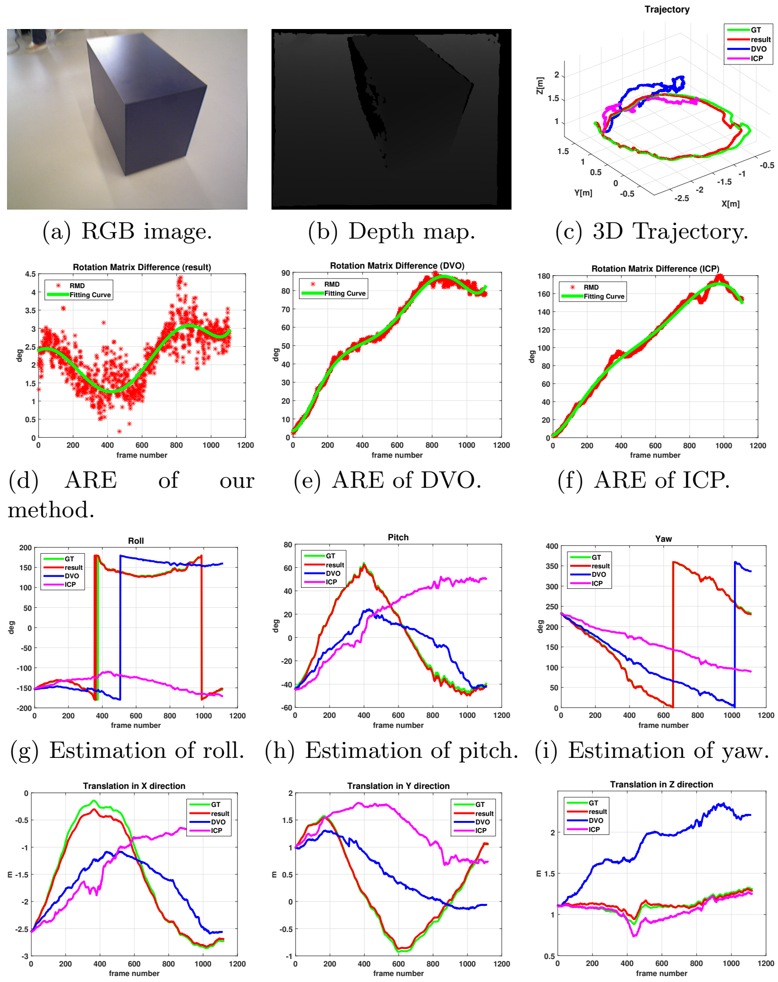
- TUM 1, 2, 3, 4, 5: fr3 (cabinet, structure_notexture/_texture _far/_near)
- IC 1,2: Living Room kt3, Office Room kt3.

Note that for TUM 4, IC 1 and IC 2, our algorithm cannot process the entire sequence due to algorithm limitations that are discussed in the following section. However, in order to remain fair, we evaluate the performance of all algorithms on the same segments of each sequence. A detailed result of the TUM 1 dataset is shown in Figure 4. We also evaluate each method using the tool given by [32] and provide root-mean-square errors $\hat{e}$ and median errors $\tilde{e}$ per second for both rotation (degree) and translation (meter) estimation in Table 1. The best performing method's error is each time indicated in bold.

It can be seen that in most cases, once the MW assumption is sufficiently met, our result provides very low drift in both rotation and translation. It is outperforming both ICP and DVO in most situations though DVO achieves better performance once there is sufficient texture in the environment. On the other hand, our method remains computationally efficient even on depth images with VGA resolution, and processes frames at about 50Hz on a CPU. While DVO is real-time capable as well (about 30 Hz), ICP quickly drops in computational efficiency as the number of points increases, and can only work in real-time with the help of a powerful GPU.

5.3 3D reconstruction

In order to demonstrate that our algorithm can work in larger scale environments such as corridors and open-space offices, we present a reconstruction result of the TAMU RGB-D dataset (corridor A const) [34] in Fig 5. The trajectory is about 40 meters long. Our algorithm robustly tracks the camera until only one dominant direction of the MW can be observed. The reconstructed structures



(a) RGB image.

(b) Depth map.

(c) 3D Trajectory.

(d) ARE of our method.

(e) ARE of DVO.

(f) ARE of ICP.

(g) Estimation of roll.

(h) Estimation of pitch.

(i) Estimation of yaw.

(j) Translation estimation in x axis.

(k) Translation estimation in y axis.

(l) Translation estimation in z axis.

Fig. 4. Evaluation of our method on the TUM dataset *cabinet* and comparison to two alternative odometry solutions (DVO and ICP). We provide the 3D trajectory, the absolution rotation error (ARE), and the translational error in each degree of freedom for each method. Our method (red curve) outperforms both DVO (blue curve) and ICP (magenta curve) in terms of absolute drift in rotation and translation. Relative pose errors can be found in Table 1. Note that only DVO uses RGB images.

(walls and ground, walls at the corridor corner) preserve orthogonality very well, which demonstrates the good quality of the motion estimation. Note that only depth information is used for the tracking. Color information is only used for visualization purposes.

5.4 Limitations and failure cases

The existence of a MW structure in the environment is key to the proposed method. Therefore, the effectiveness of our work currently has the following limitations:

- Only one mode of a MF is observed.
- If only two orthogonal planes are observed, the tracking can continue. However, due to the loss of structural information, the density distribution in the unobserved direction becomes very homogeneous, and the estimation of the respective translation becomes inaccurate.
- In the case where two MFs are very close to each other (which could happen in so-called Atlanta environments), our mean-shift scheme may converge in between the two modes, which leads to inaccurate rotation estimation and thus also potentially wrong translation estimation.

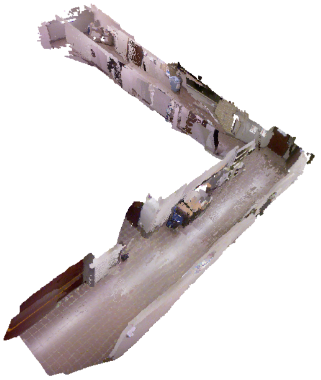


Fig. 5. Reconstruction of a corridor scene.

6 Discussion

We present an efficient alternative to the iterative closest point algorithm for real-time tracking of modern depth cameras in Manhattan Worlds. We exploit the common orthogonal structure of man-made environments in order to decouple the estimation of the rotation and the three degrees of freedom of the translation. The derived camera orientation is absolute and thus free of long-term drift, which in turn benefits the accuracy of the translation estimation as well. We achieve not only competitive accuracy, but also superior computational efficiency. Our method operates robustly in large-scale environments, even if the Manhattan World assumption is not fully met. In summary, the presented framework has high value in mobile robotics or industrial applications, where computational load or the lack of texture are major concerns. Code will be released as open-source.

Our future work consists of removing the restriction to pure Manhattan worlds. By adding a real-time mode detection and removal module, we can extend our work to the more general case of piece-wise planar environments. Interestingly, the cascaded mean-shift strategy presented in this work will still be applicable, the only difference being that the underlying manifold will no longer be $SO(3)$, but the manifold of all direction bundles with constant inscribed angles.

Acknowledgement.The research leading to these results is supported by Australian Centre for Robotic Vision. The work is furthermore supported by ARC grants DE150101365. Yi Zhou acknowledges the financial support from the China Scholarship Council for his PhD Scholarship No.201406020098.

References

1. Bachrach, A., Prentice, S., He, R., Henry, P., Huang, A.S., Krainin, M., Maturana, D., Fox, D., Roy, N.: Estimation, planning, and mapping for autonomous flight using an rgb-d camera in gps-denied environments. The International Journal of Robotics Research **31** (2012) 1320–1343
2. Bohren, J., Rusu, R.B., Jones, E.G., Marder-Eppstein, E., Pantofaru, C., Wise, M., Mösenlechner, L., Meeussen, W., Holzer, S.: Towards autonomous robotic butlers: Lessons learned with the pr2. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE (2011) 5568–5575
3. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Robotics-DL tentative, International Society for Optics and Photonics (1992) 586–606
4. Pomerleau, F., Magnenat, S., Colas, F., Liu, M., Siegwart, R.: Tracking a depth camera: Parameter exploration for fast icp. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE (2011) 3824–3829
5. Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S.: Comparing icp variants on real-world data sets. Autonomous Robots **34** (2013) 133–148
6. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on, IEEE (2011) 127–136
7. Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., McDonald, J.: Kintinuous: Spatially extended KinectFusion. In: RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras, Sydney, Australia (2012)
8. Jian, B., Vemuri, B.C.: Robust point set registration using gaussian mixture models. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33** (2011) 1633–1645
9. Chui, H., Rangarajan, A.: A new algorithm for non-rigid point matching. In: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. Volume 2., IEEE (2000) 44–51
10. Chui, H., Rangarajan, A.: A feature registration framework using mixture models. In: Mathematical Methods in Biomedical Image Analysis, 2000. Proceedings. IEEE Workshop on, IEEE (2000) 190–197
11. Tsin, Y., Kanade, T.: A correlation-based approach to robust point set registration. In: Computer Vision-ECCV 2004. Springer (2004) 558–569
12. Fitzgibbon, A.W.: Robust registration of 2d and 3d point sets. Image and Vision Computing **21** (2003) 1145–1153
13. Yang, J., Li, H., Jia, Y.: Go-icp: Solving 3d registration efficiently and globally optimally. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 1457–1464
14. Kerl, C., Sturm, J., Cremers, D.: Robust odometry estimation for rgb-d cameras. In: Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE (2013) 3748–3754
15. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2320–2327
16. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 1449–1456

17. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: Computer Vision–ECCV 2014. Springer (2014) 834–849
18. Schöps, T., Engel, J., Cremers, D.: Semi-dense visual odometry for ar on a smartphone. In: Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on, IEEE (2014) 145–150
19. Kneip, L., Zhou, Y., Li, H.: Sdicp: Semi-dense tracking based on iterative closest points. In Xianghua Xie, M.W.J., Tam, G.K.L., eds.: Proceedings of the British Machine Vision Conference (BMVC), BMVA Press (2015) 100.1–100.12
20. Weingarten, J., Siegwart, R.: 3d slam using planar segments. In: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE (2006) 3062–3067
21. Trevor, A.J., Rogers III, J.G., Christensen, H., et al.: Planar surface slam with 3d and 2d sensors. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE (2012) 3041–3048
22. Taguchi, Y., Jian, Y.D., Ramalingam, S., Feng, C.: Point-plane slam for handheld 3d sensors. In: Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE (2013) 5182–5189
23. Coughlan, J.M., Yuille, A.L.: Manhattan world: Compass direction from a single image by bayesian inference. In: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. Volume 2., IEEE (1999) 941–947
24. Košecká, J., Zhang, W.: Video compass. In: Computer VisionECCV 2002. Springer (2002) 476–490
25. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: Computer Vision–ECCV 2012. Springer (2012) 746–760
26. Straub, J., Rosman, G., Freifeld, O., Leonard, J.J., Fisher, J.W.: A mixture of manhattan frames: Beyond the manhattan world. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 3770–3777
27. Straub, J., Bhandari, N., Leonard, J.J., Fisher III, J.W.: Real-time manhattan world rotation estimation in 3d. In: IROS. (2015)
28. Fukunaga, K., Hostetler, L.D.: The estimation of the gradient of a density function, with applications in pattern recognition. Information Theory, IEEE Transactions on **21** (1975) 32–40
29. Holz, D., Holzer, S., Rusu, R.B., Behnke, S.: Real-time plane segmentation using rgb-d cameras. In: RoboCup 2011: Robot Soccer World Cup XV. Springer (2012) 306–317
30. He, K., Sun, J., Tang, X.: Guided image filtering. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35** (2013) 1397–1409
31. Carreira-Perpiñán, M.Á.: A review of mean-shift algorithms for clustering. arXiv preprint arXiv:1503.00687 (2015)
32. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2012)
33. Handa, A., Whelan, T., McDonald, J., Davison, A.J.: A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In: IEEE International Conference on Robotics and Automation (ICRA). (2014)
34. Lu, Y., Song, D.: Robustness to lighting variations: An rgb-d indoor visual odometry using line segments. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE (2015) 688–694